

## RELATÓRIO

### RELATORIO DE CRIAÇÃO UMA APLICAÇÃO WEB SOBRE UMA LOJA DE OCULOS USANDO LARAVEL

Miguel Peñaranda N°2019122

Curso Técnico Superior Profissional em  
Tecnologias e Programação de Sistemas de  
Informação

UNIDADE CURRICULAR:

Aplicações Centradas em Redes

DOCENTE:

Hugo Perdigão

# ESCOLA SUPERIOR DE TECNOLOGIAS E GESTÃO

Cofinanciado por:



REGIÃO AUTÓNOMA  
DA MADEIRA



UNIÃO EUROPEIA  
Fundo Social Europeu

## ÍNDICE

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
<b>2. DESENVOLVIMENTO DA LOJA DE OCULOS.....</b>	<b>1</b>
<b>3. MODELOS E RELAÇÕES .....</b>	<b>3</b>
<b>4. DESIÇÕES PARA O DESENVOLVIMENTO DA APLICAÇÃO.....</b>	<b>6</b>
<b>5. LOG DE ATIVIDADE .....</b>	<b>7</b>
<b>6. CONCLUSÃO.....</b>	<b>8</b>

## **AGRADECIMENTOS**

Para iniciar, quero agradecer ao docente Hugo Perdigão, que fez um grande esforço em nos ensinar um conteúdo essencial para a nossa formação em desenvolvimento web, sendo algumas destas ferramentas, jQuery, SASS y PHP. Além disso, sempre estive disposto para esclarecer as nossas dúvidas e motivar-nos a desenvolver os conhecimentos adquiridos durante as aulas.

## **1. INTRODUÇÃO**

No relatório a seguir, será apresentado o projeto proposto pelo docente Hugo Perdigão na disciplina de Aplicações Centradas em Redes. O projeto consiste em criar aplicação web sob uma loja de óculos usando Laravel, a qual permitirá aos utilizadores realizar compras e vendas de óculos.

Além disso, a aplicação web permitira fazer resenhas sobre os produtos, terão um carrinho de compras onde poderão guardar os produtos que querem comprar antes de finalizar a compra, também contarão com vista para submeter um formulário com o objetivo de pedir apoio ao administrador da aplicação. Sendo assim que o website terá uma gestão de roles de utilizadores, isto é que um utilizador que esta registado na aplicação terá acesso a outras funcionalidades que um visitante não terá e da mesma forma o administrador terá controlo sob todas as funcionalidades do website .

Enfim, o desenvolvimento desta aplicação web está feito com o objetivo de me familiarizar, aprender e melhorar os meus conhecimentos sobre php e Laravel, assim compreendendo o uso do modelo MVC que usasse em Laravel e conhecendo os comandos para a criação da base de dados, migrações, modelos, controllers e mais...

## **2. DESENVOLVIMENTO DA LOJA DE OCULOS**

Para começar o projeto consta em onze vistas, as quais usam o mesmo layout. Estas vistas são:

### **HOME**

Esta vista tem como objetivo dar uma apresentação do website ao utilizador e mostrar as últimas resenhas dos óculos para dar um pequeno feedback dos produtos que existem a vende dentro do website.

### **PRODUTOS**

A vista óculos está feita com propósito de mostrar aos utilizadores os produtos que existem á venda, podendo fazer a pesquisa dos óculos de preferência do cliente e também podendo filtrar a lista de óculos de forma ascendente ou descendente considerando o preço do mesmo.

### **DETALHES**

Esta vista apresenta todas as características do produto selecionado, uma descrição, o stock existente, o autor da publicação, o preço e um botão para adicionar o oculo ao carrinho de compras em caso de que o cliente esteja interessado. Também mostra ao utilizador as últimas resenhas exclusivas desse oculo, assim fornecendo um feedback a quem esta interessado no produto.

### **VENTA**

A vista venda tem como objetivo apresentar um formulário para submeter um oculo a venda, sendo assim que apenas pode ser enviado quem se encontra autenticado, caso contraria o utilizador deverá criar uma conta para efetuar essa ação.

### **CONTACTOS**

Esta vista tem como objetivo apresentar um formulário para submeter um pedido de apoio ao cliente, sendo assim que será visto pelo administrador na sua vista exclusiva do administrador. Alem disso, este formulário pode ser submetido por qualquer utilizador porque os problemas podem ocorrer desde o intento de criar uma conta até qualquer funcionalidade exclusiva de utilizadores autenticados.

### **LOGIN**

A vista login permite aos utilizadores acessar á sua conta, isto é autenticar-se no website permitindo-lhes ter acesso às funcionalidades exclusivas.

### **REGISTER**

A vista register permite aos utilizadores criar uma conta no website, isto é para conseguirem ter acesso às funcionalidades exclusivas da aplicação web.

### **SHOPPING CART**

Esta vista tem como objetivo apresentar o carrinho de compras unicamente se o utilizador se encontrar autenticado, desta forma podemos ver o carrinho vazio ou caso existam produtos dentro, mostra cada oculo com os seus detalhes principais e o preço total. Por tanto se o utilizador decidir comprar esses produtos tem um botão para finalizar a compra, assim concluindo a pagamento.

### **PROFILE**

A vista profile permite aos utilizadores autenticados consultar os seus dados pessoais, mudar a senha de acesso, mudar o seu nome ou email, ou eliminar a conta. Também apresenta os produtos que o atual utilizador têm a venda e se for necessário, pode mudar o preço ou repor o stock e caso já não queira vender o produto pode eliminá-lo. E por fim pode consultar o historial dos produtos em que efetuou a compra, podendo acrescentar a sua própria resenha no website sob aquele produto.

### **ADMINISTRATION**

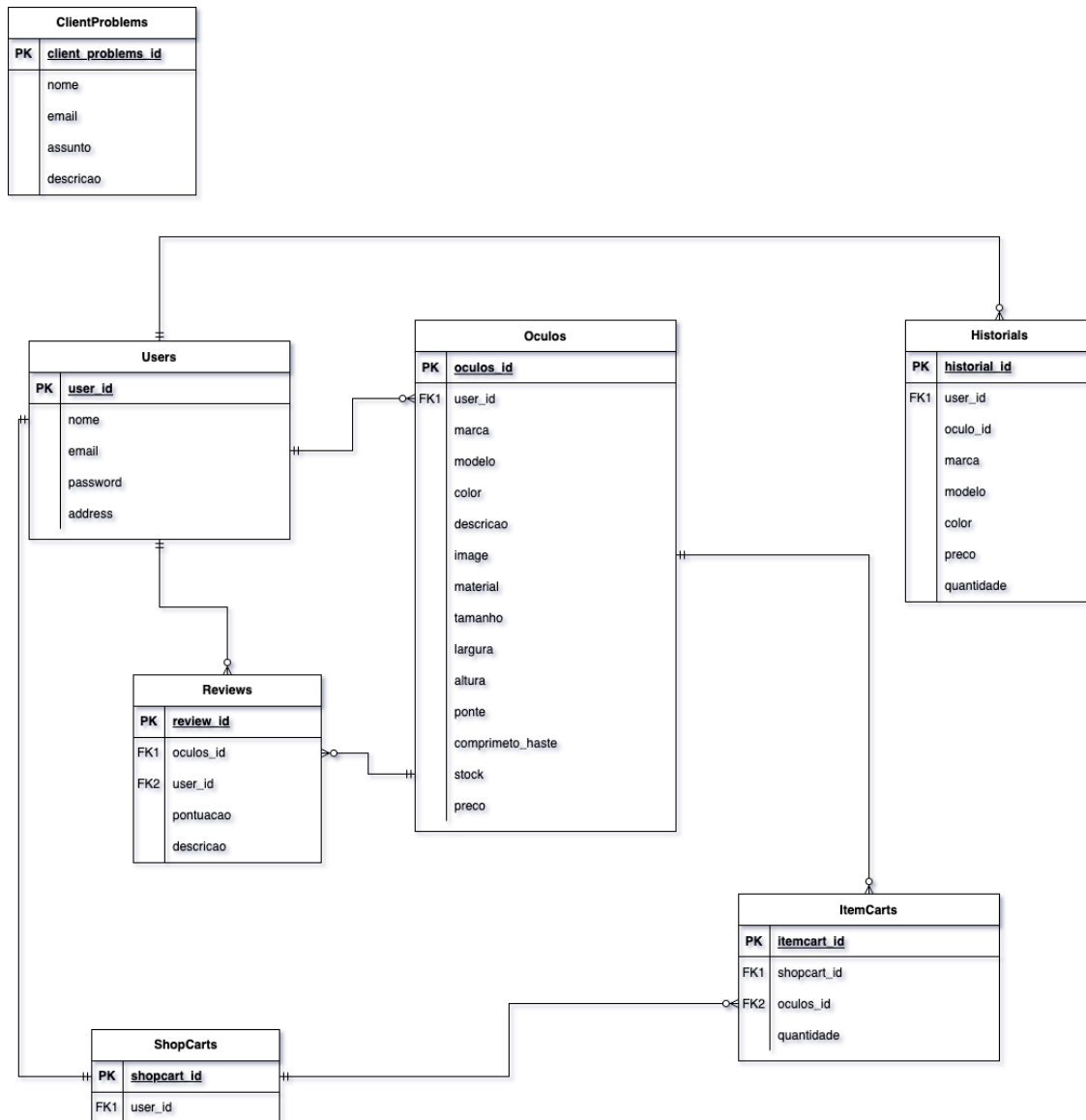
Esta vista tem como objetivo apresentar a lista de produtos existente na base de dados, para que o administrador procure em cada um, alguma característica que considere ilegal, se for o caso o administrador entra naquele oculo e pode usar um botão que só o administrador pode usar, este serve para eliminar o produto. Também apresenta uma lista de todas as solicitudes de apoio requisitadas para que o mesmo se punha em contacto com os clientes e possa brindar-lhe a sua ajuda.

### **ERRO**

A vista de erro tem apresenta uma mensagem erro 404 e um botão que redirige ao utilizador a vista home. Desde modo, esta vista só será visível quando o cliente navegar numa rota incorreta.

### 3. MODELOS E RELAÇÕES

Para o desenvolvimento da aplicação web implementei uma base de dados relacional com sete tabelas, deste modo elaborei um modelo inicial do desenho de uma notação Crow's Foot para conseguir interligar os modelos de forma logical e funcional.



*Figura 1- Notação Crow's Foot*

## **MODELO – USER**

O modelo User tem como objetivo armazenar a informação dos utilizadores, se for o caso pode guardar o token da sessão para relembrar o usuário quando entrar no website e também foi lhe adicionado um novo campo “isAdmin” que permite diferenciar o administrador dos clientes.

Esta tabela tem relação de um para muitos com os modelos Review, Oculo e Historial, porque:

- Um utilizador pode ter uma ou muitas resenha
- Um utilizador pode ter um ou muitos óculos á venda
- Um utilizador pode ter um ou muitos histórias, ou seja, histórico de compras

## **MODELO – REVIEW**

O modelo Review tem como objetivo armazenar a informação sob as resenhas submetidas pelos clientes da aplicação.

Esta tabela tem relação muitos para um com os modelos User e Oculo, porque:

- Muitas resenhas podem pertencer a um único utilizador
- Muitas resenhas podem pertencer a um único oculo

## **MODELO – OCULO**

O modelo Oculo tem como objetivo armazenar a informação sob os óculos publicados pelos clientes da aplicação.

Esta tabela tem relações com os modelos User, Review e ItemsCart, porque:

- Muitos óculos podem pertencer a um único utilizador
- Um oculo pode ter muitas resenhas
- Um oculo pode pertencer a muitos carrinhos de compras

## **MODELO – HISTORIAL**

O modelo Historial tem como objetivo armazenar a informação sob as compras emitidas pelos clientes da aplicação, permitindo-lhes a partir desta emitir resenhas após testar o produto.

Esta tabela tem relação de muitos para um com o modelo User porque:

- Muitas compras podem ser feitas por um único utilizador



### **MODELO – SHOPCART**

O modelo ShopCart tem como objetivo associar um carrinho de compras á um utilizador, quando o mesmo regista-se na aplicação.

Esta tabela tem uma relação de um para um com o modelo User porque:

- Um utilizador só pode ter um único carrinho de compras associado

### **MODELO – ITEMSCART**

O modelo ItemsCart tem como objetivo armazenar a informação sob os produtos que foram adicionar ao carrinho de compras de um utilizador.

Esta tabela tem relação muitos para um com os modelos ShoppingCart e Oculo, porque:

- Muitos produtos do carrinho podem ter um oculo
- Muitos produtos do carrinho podem ter unicamente um carrinho de compras

### **MODELO – ClientProblem**

O modelo ClientProblem tem como objetivo armazenar a informação sob os problemas enviados ao gerente da aplicação para que o mesmo possa brindar apoio ao cliente.

Esta tabela não tem relações porque foi considerado que os problemas podem abranger desde os utilizadores que não se encontram registados até casos mais específicos.

#### **4. DESIÇÕES PARA O DESENVOLVIMENTO DA APLICAÇÃO**

No layout e algumas vistas restringi alguns botões para quem não se encontra autenticado e outros para quem não é administrador. Desta forma, consegue-se limitar a capacidade do clientes casuais e se eles quiseres trabalhar com a aplicação, estes são obrigados a fazer o registo.

Na vista que lista os óculos os clientes tem a possibilidade de procurar através de uma barra de navegação em que implementei um endpoint que procura na tabela tendo em conta os campos: marca e modelo. Também existem dois botões que permitem mostrar a lista de duas formas, ordenada de forma ascendente por preço ou descendente por preço.

Na vista de detalhes de um produto existem três botões um unicamente para que o administrador poder eliminar o mesmo se for necessário, outro para o clientes que não se encontram autenticados e quiserem comprar o produto, os redirige à vista de register e por último um botão que abre um modal com um formulário de a quantidade a comprar, se, se a quantidade ultrapassar o stock existente é enviado uma mensagem de erro e se o produto já existir no carrinho apenas é modificado a quantidade de aquele que esta no carrinho, sendo assim enviado ao carrinho.

Na vista que permite vender os produtos, o endpoint que importa o produto na tabela de oculo tem validações para todos os campos e restringe o máximo de stock que pode ser inserido por o utilizador.

Na vista de apoio ao cliente, o endpoint que importa o produto na tabela de ClientProblem tem oculo tem validações para todos os campos.

No vista do shoppingcart, se houver produtos existe um botão para eliminar o produto do carrinho, o endpoint para este foi implementado de forma que se houver mais do que um produto apenas é reduzida a quantidade do mesmo, mas se houver apenas um então é eliminado completamente o card do carrinho. Também tem o botão para finalizar a compra que através de verificações, tais como se nesse momento existe mais stock do que a quantidade inserida, então confirma a compra, caso contrário envia uma mensagem de erro indicando o problema.

## **5. LOG DE ATIVIDADE**

### **Sexta-Feira, 03 de Novembro de 2023**

- Inicialização de Projeto em Laravel
- Configurar o Projeto para usar bootstrap, sass e um reste
- Criação de um Layout

### **Terça-Feira, 21 de Novembro de 2023**

- Implementação das views, routes e sass
- Desenvolvimento das views

### **Quarta-Feira, 29 de Novembro de 2023**

- Criação da base de dados
- Implementação do Laravel Breeze
- Perceber bem o funcionamento do Laravel Breeze
- Adaptação das minhas views para conseguir fazer o registo, login e updates no profile

### **Sexta-Feira, 1 de Dezembro de 2023**

- Criação de um modelo CROWS FOOT sobre como iria ser a minha base de dados
- Listar quais são os endpoints que sabia que iam ser necesarios
- Implementação de todas as tabelas e relações (migrations e models)
- Implementação dos controllers dos modelos em que são necessários
- Implementação de alguns métodos e endpoints
- Adicionar ao user um campo booleano "isAdmin" para definir que é o administrador
- Configurar as views para restringir o acesso a algumas vistas ou componentes que só podem ser usadas pelo administrador
- Configurar as views para restringir o acesso a algumas vistas ou componentes que só podem ser usadas por utilizador autenticados na aplicação web
- Configurar o projeto para que um utilizador navegar numa rota que não existe redirecionar-la para a View de Erro

### **Sábado, 2 de Dezembro de 2023**

- Implementação de todos os métodos e endpoints restantes para o funcionamento da aplicação
- Implementação de validações aos campos nos post de formularios
- Correção de alguns erros
- Melhorar alguns métodos e views

## **6. CONCLUSÃO**

Em conclusão, a aplicação web desenvolvida neste trabalho, além de demonstrar como as habilidades e conhecimentos adquiridos na Unidade Curricular Aplicações Centradas em Redes foram aplicados para a elaboração de uma loja de óculos em Laravel, também contribuiu para reforçar o aprendizado nas aulas e proporcionou uma oportunidade para pôr em prática as habilidades adquiridas ao longo das lições.

Por tanto realização de uma loja de óculos em Laravel a foi um grande desafio devido à sua complexidade. No entanto, foi muito valiosa, pois permitiu aprender a encontrar soluções para os problemas apresentados e procurar métodos eficazes para cumprir todas as funções que a aplicação web devia ter.

Enfim, para isto foram utilizadas as ferramentas Laravel, Laravel Breeze, Bootstrap, JavaScript, sendo assim um projeto bastante completo enriquecedor a nível de conhecimentos. De modo geral, este projeto é um exemplo eficaz de como os conceitos aprendidos podem ser aplicados para resolver problemas práticos e reais. O projeto resultou numa experiência que ajudou a desenvolver habilidades importantes para obter um melhor desenvolvimento no mundo de aplicações web.